

시험에  
나오는 것만  
공부한다!



보기만 달리하여

종류나 순서를 묻는 114문제  
정보처리기사 필기



1. 소프트웨어 공학의 기본 원칙이라고 볼 수 없는 것은?

- ① 품질 높은 소프트웨어 상품 개발
- ② 지속적인 검증 시행
- ③ 결과에 대한 명확한 기록 유지
- ④ 최대한 많은 인력 투입

[해설]

소프트웨어 공학의 기본 원칙

- 품질 높은 소프트웨어 상품 개발
- 지속적인 검증 시행
- 결과에 대한 명확한 기록 유지
- 효율적인 인력 투입

2. 시스템의 구성 요소로 볼 수 없는 것은?

- ① Process
- ② Feedback
- ③ Maintenance
- ④ Control

[해설]

시스템의 구성 요소

- 입력(Input)
- 처리(Process)
- 출력(Output)
- 제어(Control)
- 피드백(Feedback)

3. 요구사항 정의 및 분석·설계의 결과물을 표현하기 위한 모델링 과정에서 사용되는 다이어그램(Diagram)이 아닌 것은?

- ① Data Flow Diagram
- ② UML Diagram
- ③ E-R Diagram
- ④ AVL Diagram

[해설]

요구사항 정의, 분석, 설계에 사용되는 다이어그램

- 자료 흐름도(DFD; Data Flow Diagram)
- UML 다이어그램
- E-R(개체-관계) 다이어그램

4. 현행 시스템 분석에서 고려하지 않아도 되는 항목은?

- ① DBMS 분석
- ② 네트워크 분석
- ③ 운영체제 분석
- ④ 인적 자원 분석

[해설]

현행 시스템 파악 절차

- 시스템 구성 파악
- 시스템 기능 파악
- 시스템 인터페이스 파악
- 아키텍처 구성 파악
- 소프트웨어(DBMS, 운영체제 등) 구성 파악
- 하드웨어 구성 파악
- 네트워크 구성 파악

5. GoF(Gang of Four) 디자인 패턴을 생성, 구조, 행동 패턴의 세 그룹으로 분류할 때, 구조 패턴이 아닌 것은?

- ① Adapter 패턴
- ② Bridge 패턴
- ③ Builder 패턴
- ④ Proxy 패턴

[해설]

구조 패턴의 종류

- 어댑터(Adapter)
- 브리지(Bridge)
- 컴포지트(Composite)
- 데코레이터(Decorator)
- 퍼싸드(Facade)
- 플라이웨이트(Flyweight)
- 프록시(Proxy)

6. 객체 지향 소프트웨어 설계 시 디자인 패턴을 구성하는 요소로서 가장 거리가 먼 것은?

- ① 개발자 이름
- ② 문제 및 배경
- ③ 사례
- ④ 샘플 코드

[해설]

디자인 패턴의 구성 요소

- 문제 및 배경
- 실제 적용된 사례
- 재사용이 가능한 샘플 코드

7. 소프트웨어 설계에서 사용되는 대표적인 추상화(Abstraction) 기법이 아닌 것은?

- ① 자료 추상화
- ② 제어 추상화
- ③ 과정 추상화
- ④ 강도 추상화

[해설]

추상화 유형

- 과정 추상화
- 데이터(자료) 추상화
- 제어 추상화

8. UML에서 시퀀스 다이어그램의 구성 항목에 해당하지 않는 것은?

- ① 생명선
- ② 실행
- ③ 확장
- ④ 메시지

[해설]

순차 다이어그램의 구성 요소

- 액터(Actor)
- 객체(Object)
- 생명선(Lifeline)
- 실행 상자(Active Box)
- 메시지(Message)

9. 디자인 패턴 중에서 행위적 패턴에 속하지 않는 것은?

- ① 커맨드(Command) 패턴
- ② 옵저버(Observer) 패턴
- ③ 프로토타입(Prototype) 패턴
- ④ 상태(State) 패턴

[해설]

행위 패턴의 종류

- 책임 연쇄(Chain of Responsibility)
- 커맨드(Command)
- 인터프리터(Interpreter)
- 반복자(Iterator)
- 중재자(Mediator)
- 메멘토(Memento)
- 옵저버(Observer)
- 상태(State)
- 전략(Strategy)
- 템플릿 메소드(Template Method)
- 방문자(Visitor)

## 10. UML 다이어그램이 아닌 것은?

- ① 액티비티 다이어그램(Activity Diagram)
- ② 절차 다이어그램(Procedural Diagram)
- ③ 클래스 다이어그램(Class Diagram)
- ④ 시퀀스 다이어그램(Sequence Diagram)

### [해설]

#### UML 다이어그램의 종류

- 구조적(Structural) 다이어그램 - 정적 모델링
  - 클래스 다이어그램(Class Diagram)
  - 객체 다이어그램(Object Diagram)
  - 컴포넌트 다이어그램(Component Diagram)
  - 배치 다이어그램(Deployment Diagram)
  - 복합체 구조 다이어그램(Composite Structure Diagram)
  - 패키지 다이어그램(Package Diagram)
- 행위(Behavioral) 다이어그램 - 동적 모델링
  - 유스케이스 다이어그램(Use Case Diagram)
  - 순차 다이어그램(Sequence Diagram)
  - 커뮤니케이션 다이어그램(Communication Diagram)
  - 상태 다이어그램(State Diagram)
  - 활동 다이어그램(Activity Diagram)
  - 상호작용 개요 다이어그램(Interaction Overview Diagram)
  - 타이밍 다이어그램(Timing Diagram)

## 11. 소프트웨어 품질 측정을 위해 개발자 관점에서 고려해야 할 항목으로 거리가 먼 것은?

- ① 정확성
- ② 무결성
- ③ 사용성
- ④ 간결성

### [해설]

#### 소프트웨어의 품질 특성

- 기능성 : 적합성, 정확성, 상호 운용성, 보안성, 호환성
- 신뢰성 : 성숙성, 고장 허용성, 회복성
- 사용성 : 이해성, 학습성, 운용성, 친밀성
- 효율성 : 시간 효율성, 자원 효율성
- 유지 보수성 : 분석성, 변경성, 안정성, 시험성
- 이식성 : 적용성, 설치성, 대체성, 공존성
- 호환성 : 공존성, 상호 운영성
- 보안성 : 기밀성, 무결성, 부인방지, 책임추적성, 인증성

12. UML 다이어그램 중 정적 다이어그램이 아닌 것은?

- ① 컴포넌트 다이어그램
- ② 배치 다이어그램
- ③ 순차 다이어그램
- ④ 패키지 다이어그램

[해설]

구조적(Structural) 다이어그램 - 정적 모델링

- 클래스 다이어그램(Class Diagram)
- 객체 다이어그램(Object Diagram)
- 컴포넌트 다이어그램(Component Diagram)
- 배치 다이어그램(Deployment Diagram)
- 복합체 구조 다이어그램(Composite Structure Diagram)
- 패키지 다이어그램(Package Diagram)

13. UML 모델에서 사용하는 Structural Diagram에 속하지 않은 것은?

- ① Class Diagram
- ② Object Diagram
- ③ Component Diagram
- ④ Activity Diagram

[해설]

UML 다이어그램의 종류

구조적(Structural) 다이어그램 - 정적 모델링

- 클래스 다이어그램(Class Diagram)
- 객체 다이어그램(Object Diagram)
- 컴포넌트 다이어그램(Component Diagram)
- 배치 다이어그램(Deployment Diagram)
- 복합체 구조 다이어그램(Composite Structure Diagram)
- 패키지 다이어그램(Package Diagram)

14. UML에서 활용되는 다이어그램 중, 시스템의 동작을 표현하는 행위(Behavioral) 다이어그램에 해당하지 않는 것은?

- ① 유스케이스 다이어그램(Use Case Diagram)
- ② 시퀀스 다이어그램(Sequence Diagram)
- ③ 활동 다이어그램(Activity Diagram)
- ④ 배치 다이어그램(Deployment Diagram)

[해설]

행위(Behavioral) 다이어그램 - 동적 모델링

- 유스케이스 다이어그램(Use Case Diagram)
- 순차 다이어그램(Sequence Diagram)
- 커뮤니케이션 다이어그램(Communication Diagram)
- 상태 다이어그램(State Diagram)
- 활동 다이어그램(Activity Diagram)
- 상호작용 개요 다이어그램(Interaction Overview Diagram)
- 타이밍 다이어그램(Timing Diagram)

15. ISO/IEC 9126의 소프트웨어 품질 특성 중 기능성(Functionality)의 하위 특성으로 옳지 않은 것은?

- ① 학습성
- ② 적합성
- ③ 정확성
- ④ 보안성

[해설]

ISO/IEC 9126의 소프트웨어 품질 - 기능성

적합성, 정확성, 상호 운용성, 보안성, 호환성

16. 애자일 방법론에 해당하지 않는 것은?

- ① 기능 중심 개발
- ② 스크럼
- ③ 익스트림 프로그래밍
- ④ 모듈 중심 개발

[해설]

애자일 모형을 기반으로 하는 소프트웨어 개발 모형

스크럼, XP(eXtreme Programming), 칸반, Lean, 크리스탈, ASD, 기능 중심 개발(FDD), DSDM, DAD 등

17. 다음 중 상위 CASE 도구가 지원하는 주요 기능으로 볼 수 없는 것은?

- ① 모델들 사이의 모순 검사 기능
- ② 전체 소스 코드 생성 기능
- ③ 모델의 오류 검증 기능
- ④ 자료 흐름도 작성 기능

[해설]

상위 CASE 도구가 지원하는 주요 기능

- 모델들 사이의 모순 검사 기능
- 모델의 오류 검증 기능
- 자료 흐름도 작성 기능

18. 소프트웨어 아키텍처 설계에서 시스템 품질 속성이 아닌 것은?

- ① 가용성(Availability)
- ② 독립성(Isolation)
- ③ 변경 용이성(Modifiability)
- ④ 사용성(Usability)

[해설]

소프트웨어 아키텍처의 시스템 품질 속성

- 시스템 측면 : 성능, 보안, 가용성, 기능성, 사용성, 변경 용이성, 확장성 등
- 비즈니스 측면 : 시장 적시성, 비용과 혜택, 예상 시스템 수명 등
- 아키텍처 측면 : 개념적 무결성, 정확성, 완결성, 구축 가능성 등

19. GoF(Gangs of Four) 디자인 패턴 분류에 해당하지 않는 것은?

- ① 생성 패턴
- ② 구조 패턴
- ③ 행위 패턴
- ④ 추상 패턴

[해설]

디자인 패턴 분류

- 생성 패턴(Creational Pattern)
- 구조 패턴(Structural Pattern)
- 행위 패턴(Behavioral Pattern)

20. CASE가 갖고 있는 주요 기능이 아닌 것은?

- ① 그래픽 지원
- ② 소프트웨어 생명 주기 전 단계의 연결
- ③ 언어 번역
- ④ 다양한 소프트웨어 개발 모형 지원

[해설]

CASE의 주요 기능

- S/W 라이프 사이클 전 단계의 연결
- 모델들의 모순 검사
- 모델들의 오류 검증
- 자료 흐름도(DFD) 등의 다이어그램(Diagram) 작성
- 다양한 소프트웨어 개발 모형 지원
- 시스템 문서화 및 명세화를 위한 그래픽 지원

21. XP(eXtreme Programming)의 기본 원리로 볼 수 없는 것은?

- ① Linear Sequential Method
- ② Pair Programming
- ③ Collective Ownership
- ④ Continuous Integration

[해설]

XP의 주요 실천 방법(Practice)

- Pair Programming(짝 프로그래밍)
- Collective Ownership(공동 코드 소유)
- Test-Driven Development(테스트 주도 개발)
- Whole Team(전체 팀)
- Continuous Integration(계속적인 통합)
- Design Improvement(디자인 개선)
- Refactoring(리팩토링)
- Small Releases(소규모 릴리즈)

22. 다음 중 요구사항 모델링에 활용되지 않는 것은?

- ① 애자일(Agile) 방법
- ② 유스케이스 다이어그램(Use Case Diagram)
- ③ 시퀀스 다이어그램(Sequence Diagram)
- ④ 단계 다이어그램(Phase Diagram)

[해설]

요구사항 모델링 활용

- 애자일(Agile) 방법
- 유스케이스 다이어그램(Use Case Diagram)
- 순차 다이어그램(Sequence Diagram)

23. 유스케이스(Use Case)의 구성 요소 간의 관계에 포함되지 않는 것은?

- ① 연관
- ② 확장
- ③ 구체화
- ④ 일반화

[해설]

유스케이스 다이어그램 구성 요소 간의 관계

연관 관계, 포함 관계, 확장 관계, 일반화 관계

24. 코드의 기본 기능으로 거리가 먼 것은?

- ① 복잡성
- ② 표준화
- ③ 분류
- ④ 식별

[해설]

코드의 기능

식별 기능, 분류 기능, 배열 기능, 표준화 기능, 간소화 기능

25. GoF(Gangs of Four) 디자인 패턴 중 생성 패턴으로 옳은 것은?

- ① Singleton Pattern
- ② Adapter Pattern
- ③ Decorator Pattern
- ④ State Pattern

[해설]

생성 패턴의 종류

- 추상 팩토리(Abstract Factory)
- 빌더(Builder)
- 팩토리 메소드(Factory Method)
- 프로토타입(Prototype)
- 싱글톤(Singleton)



26. 다음 중 SOLID 원칙이라고 불리는 객체지향 설계 원칙에 속하지 않는 것은?

- ① ISP(Interface Segregation Principle)
- ② DIP(Dependency Inversion Principle)
- ③ LSP(Liskov Substitution Principle)
- ④ SSO(Single Sign On)

[해설]

객체지향 설계 원칙

- 단일 책임 원칙(SRP; Single Responsibility Principle)
- 개방-폐쇄 원칙(OCF; Open-Closed Principle)
- 리스코프 치환 원칙(LSP; Liskov Substitution Principle)
- 인터페이스 분리 원칙(ISP; Interface Segregation Principle)
- 의존 역전 원칙(DIP; Dependency Inversion Principle)

27. 미들웨어 솔루션의 유형에 포함되지 않는 것은?

- ① WAS
- ② Web Server
- ③ RPC
- ④ ORB

[해설]

미들웨어의 종류

DB, RPC, MOM, TP-Monitor, ORB, WAS 등

28. 럼바우(Rumbaugh)의 객체 지향 분석에서 사용하는 분석 활동으로 옳은 것은?

- ① 객체 모델링, 동적 모델링, 정적 모델링
- ② 객체 모델링, 동적 모델링, 기능 모델링
- ③ 동적 모델링, 기능 모델링, 정적 모델링
- ④ 정적 모델링, 객체 모델링, 기능 모델링

[해설]

럼바우의 분석 기법

- 객체 모델링(Object Modeling)
- 동적 모델링(Dynamic Modeling)
- 기능 모델링(Functional Modeling)

29. 모바일 기기에서 사용하는 NUI 인터페이스에 속하지 않는 것은 무엇인가?

- ① Pinch
- ② Press
- ③ Flow
- ④ Flick

[해설]

주요 모바일 제스처

Tap, Double Tap, Drag, Pan, Press, Flick, Pinch 등

30. DBMS 분석 시 고려사항으로 거리가 먼 것은?

- ① 가용성
- ② 성능
- ③ 네트워크 구성도
- ④ 상호 호환성

[해설]

DBMS 분석 시 고려사항

가용성, 성능, 기술 지원, 상호 호환성, 구축 비용

31. UML의 기본 구성 요소가 아닌 것은?

- ① Things
- ② Terminal
- ③ Relationship
- ④ Diagram

[해설]

UML의 구성 요소

- 사물(Things)
- 관계(Relationships)
- 다이어그램(Diagram)

32. 요구사항 분석 시에 필요한 기술로 가장 거리가 먼 것은?

- ① 청취와 인터뷰 질문 기술
- ② 분석과 중재 기술
- ③ 설계 및 코딩 기술
- ④ 관찰 및 모델 작성 기술

[해설]

요구사항 분석 단계에서 사용되는 기술

- 청취와 인터뷰 질문 기술
- 분석과 중재 기술
- 관찰 및 모델 작성 기술

33. 소프트웨어의 상위 설계에 속하지 않는 것은?

- ① 아키텍처 설계
- ② 모듈 설계
- ③ 인터페이스 정의
- ④ 사용자 인터페이스 설계

[해설]

소프트웨어의 설계

- 상위 설계(아키텍처 설계) : 구조 설계, DB 설계, 인터페이스 설계
- 하위 설계(모듈 설계) : 컴포넌트 설계, 자료 구조 설계, 알고리즘 설계

34. CASE(Computer-Aided Software Engineering)의 원천 기술이 아닌 것은?

- ① 구조적 기법
- ② 프로토타이핑 기술
- ③ 정보 저장소 기술
- ④ 일괄 처리 기술

[해설]

CASE의 원천 기술

- 구조적 기법
- 프로토타이핑 기술
- 자동 프로그래밍 기술
- 정보 저장소 기술
- 분산 처리 기술

35. 소프트웨어의 사용자 인터페이스 개발 시스템(User Interface Development System)이 가져야 할 기능이 아닌 것은?

- ① 사용자 입력의 검증
- ② 에러 처리와 에러 메시지 처리
- ③ 도움과 프롬프트(Prompt) 제공
- ④ 소스 코드 분석 및 오류 복구

[해설]

사용자 인터페이스 개발 시스템의 기능

입력 검증, 에러 처리, 도움 제공

36. 익스트림 프로그래밍(eXtreme Programming)의 5가지 가치에 속하지 않는 것은?

- ① 의사소통
- ② 단순성
- ③ 피드백
- ④ 고객 배제

[해설]

XP(eXtreme Programming)의 5가지 핵심 가치

- 의사소통(Communication)
- 단순성(Simplicity)
- 용기(Courage)
- 존중(Respect)
- 피드백(Feedback)

37. 자료 흐름도(DFD)의 각 요소별 표기 형태의 연결이 옳지 않은 것은?

- ① Process : 원
- ② Data Flow : 화살표
- ③ Data Store : 삼각형
- ④ Terminator : 사각형

[해설]

자료 흐름도(DFD) 표기법

- 프로세스(Process) : 원
- 자료 흐름(Data Flow) : 화살표
- 자료 저장소(Data Store) : 평행선
- 단말(Terminator) : 사각형

38. 소프트웨어 설계 시 구축된 플랫폼의 성능 특성 분석에 사용되는 측정 항목이 아닌 것은?

- ① 응답시간(Response Time)
- ② 가용성(Availability)
- ③ 사용률(Utilization)
- ④ 서버 튜닝(Server Tuning)

[해설]

플랫폼의 성능 특성 분석에 사용되는 측정 항목

- 응답시간(Response Time)
- 가용성(Availability)
- 사용률(Utilization)

39. 소프트웨어를 재사용함으로써 얻을 수 있는 이점으로 가장 거리가 먼 것은?

- ① 생산성 증가
- ② 프로젝트 문서 공유
- ③ 소프트웨어 품질 향상
- ④ 새로운 개발 방법론 도입 용이

[해설]

소프트웨어 재사용의 장점

- 개발 시간과 비용 단축
- 소프트웨어 품질 향상
- 소프트웨어 개발의 생산성 향상
- 프로젝트 실패의 위험 감소
- 시스템 구축 방법에 대한 지식 공유
- 시스템 명세, 설계, 코드 등 문서 공유

40. 소프트웨어 설치 매뉴얼에 포함될 항목이 아닌 것은?

- ① 제품 소프트웨어 개요
- ② 설치 관련 파일
- ③ 프로그램 삭제
- ④ 소프트웨어 개발 기간

[해설]

소프트웨어 설치 매뉴얼의 기본 항목

- 소프트웨어 개요
- 설치 관련 파일
- 설치 아이콘(Installation)
- 프로그램 삭제
- 관련 추가 정보

41. 다음 중 블랙박스 검사 기법은?

- ① 경계값 분석
- ② 조건 검사
- ③ 기초 경로 검사
- ④ 루프 검사

[해설]

- 화이트박스 테스트 : 기초 경로 검사, 제어 구조 검사(조건 검사, 루프 검사, 데이터 흐름 검사) 등
- 블랙박스 테스트 : 동치 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등

42. 화이트박스 검사 기법에 해당하는 것으로만 짝지어진 것은?

- ㉠ 데이터 흐름 검사
- ㉡ 루프 검사
- ㉢ 동등 분할 검사
- ㉣ 경계값 분석
- ㉤ 원인 결과 그래프 기법
- ㉥ 오류 예측 기법

- ① ㉠, ㉡
- ② ㉠, ㉣
- ③ ㉡, ㉤
- ④ ㉢, ㉥

[해설]

- 화이트박스 테스트 : 기초 경로 검사, 제어 구조 검사(조건 검사, 루프 검사, 데이터 흐름 검사) 등
- 블랙박스 테스트 : 동치 분할 검사, 경계값 분석, 원인-효과 그래프 검사, 오류 예측 검사, 비교 검사 등

43. 형상 관리 도구의 주요 기능으로 거리가 먼 것은?

- ① 정규화(Normalization)
- ② 체크인(Check-in)
- ③ 체크아웃(Check-out)
- ④ 커밋(Commit)

[해설]

소프트웨어의 버전 등록 관련 주요 기능

- 저장소(Repository)
- 가져오기(Import)
- 체크아웃(Check-Out)
- 체크인(Check-In)
- 커밋(Commit)
- 동기화(Update)

44. 반정규화(Denormalization) 유형 중 중복 테이블을 추가하는 방법에 해당하지 않는 것은?

- ① 빌드 테이블의 추가
- ② 집계 테이블의 추가
- ③ 진행 테이블의 추가
- ④ 특정 부분만을 포함하는 테이블 추가

[해설]

중복 테이블 추가 방법

- 집계 테이블의 추가
- 진행 테이블의 추가
- 특정 부분만을 포함하는 테이블의 추가

45. 테스트 케이스 자동 생성 도구를 이용하여 테스트 데이터를 찾아내는 방법이 아닌 것은?

- ① 스텝(Stub)과 드라이버(Driver)
- ② 입력 도메인 분석
- ③ 랜덤(Random) 테스트
- ④ 자료 흐름도

[해설]

테스트 케이스 생성 도구


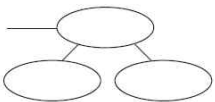
- 자료 흐름도
- 기능 테스트
- 입력 도메인 분석
- 랜덤 테스트

46. E-R 다이어그램의 표기법으로 옳지 않은 것은?

- ① 개체 타입 - 사각형
- ② 속성 - 타원
- ③ 관계 집합 - 삼각형
- ④ 개체 타입과 속성을 연결 - 선

[해설]

E-R 다이어그램

기호	기호 이름	의미
	사각형	개체(Entity) 타입
	마름모	관계(Relationship) 타입
	타원	속성(Attribute)
	이중 타원	다중값 속성(복합 속성)
	밑줄 타원	기본키 속성
	복수 타원	복합 속성
	선, 링크	개체 타입과 속성을 연결

47. 스택(STACK)의 응용 분야로 거리가 먼 것은?

- ① 인터럽트의 처리
- ② 주식의 계산
- ③ 서브루틴의 복귀 번지 저장
- ④ 운영체제의 작업 스케줄링

[해설]

스택의 응용 분야

- 함수 호출의 순서 제어
- 인터럽트의 처리
- 주식 계산 및 주식 표기법
- 컴파일러를 이용한 언어 번역
- 부 프로그램 호출 시 복귀주소 저장
- 서브루틴 호출 및 복귀 주소 저장

48. 다음 중 스택을 이용한 연산과 거리가 먼 것은?

- ① 선택 정렬
- ② 재귀 호출
- ③ 후위 표현(Post-Fix Expression)의 연산
- ④ 깊이 우선 탐색

[해설]

스택(Stack)을 이용한 연산은 '재귀 호출, 후위(Postfix) 표기법, 깊이 우선 탐색'과 같이 왔던 길을 되돌아가는 경우에 사용됩니다.

49. 다음 중 선형 구조로만 묶인 것은?

- ① 스택, 트리
- ② 큐, 데크
- ③ 큐, 그래프
- ④ 리스트, 그래프

[해설]

자료 구조 분류

- 선형 구조 : 배열, 선형 리스트, 스택, 큐, 데크
- 비선형 구조 : 트리, 그래프

50. 소스 코드 품질 분석 도구 중 정적 분석 도구가 아닌 것은?

- ① pmd
- ② checkstyle
- ③ valMeter
- ④ cppcheck

[해설]

소스 코드 품질 분석 도구

- 정적 분석 도구 : pmd, cppcheck, SonarQube, checkstyle, ccm, cobertura 등
- 동적 분석 도구 : Avalanche, Valgrind 등

51. 인터페이스 보안을 위해 네트워크 영역에 적용될 수 있는 것으로 거리가 먼 것은?

- ① IPSec
- ② SSL
- ③ SMTP
- ④ S-HTTP

[해설]

인터페이스 보안 - 네트워크 영역

IPSec, SSL, S-HTTP 등

52. 다음 중 단위 테스트 도구로 사용될 수 없는 것은?

- ① CppUnit
- ② JUnit
- ③ HttpUnit
- ④ IgpUnit

[해설]

xUnit의 종류

JUnit, CppUnit, NUnit, HttpUnit 등



53. 정보 시스템 개발 단계에서 프로그래밍 언어 선택 시 고려할 사항으로 가장 거리가 먼 것은?

- ① 개발 정보 시스템의 특성
- ② 사용자의 요구사항
- ③ 컴파일러의 가용성
- ④ 컴파일러의 독창성

[해설]

프로그래밍 언어의 선정 기준

- 친밀감
- 언어의 능력
- 처리의 효율성
- 프로그램 구조
- 프로그램의 길이
- 이식성
- 과거의 개발 실적
- 알고리즘과 계산상의 난이도
- 자료 구조의 난이도
- 성능 고려 사항들
- 대상 업무의 성격
- 소프트웨어의 수행 환경
- 개발 담당자의 경험과 지식
- 사용자의 요구사항
- 컴파일러의 이용 가능성

54. 분산 데이터베이스의 투명성(Transparency)에 해당하지 않는 것은?

- ① Location Transparency
- ② Replication Transparency
- ③ Failure Transparency
- ④ Media Access Transparency

[해설]

분산 데이터베이스의 목표

- 위치 투명성(Location Transparency)
- 중복 투명성(Replication Transparency)
- 병행 투명성(Concurrency Transparency)
- 장애 투명성(Failure Transparency)

55. 소프트웨어 형상 관리에서 관리 항목에 포함되지 않는 것은?

- ① 프로젝트 요구 분석서
- ② 소스 코드
- ③ 운영 및 설치 지침서
- ④ 프로젝트 개발 비용

[해설]

소프트웨어 형상 관리의 관리 항목

소스 코드, 분석서, 지침서, 설계서, 프로젝트 계획, 프로그램, 테스트 케이스 등

56. 병렬 데이터베이스 환경 중 수평 분할에서 활용되는 분할 기법이 아닌 것은?

- ① 라운드-로빈
- ② 범위 분할
- ③ 예측 분할
- ④ 해시 분할

[해설]

파티션의 종류

- 범위 분할(Range Partitioning)
- 해시 분할(Hash Partitioning)
- 조합 분할(Composite Partitioning)
- 목록 분할(List Partitioning)
- 라운드 로빈 분할(Round Robin Partitioning)

57. 알고리즘 설계 기법으로 거리가 먼 것은?

- ① Divide and Conquer
- ② Greedy
- ③ Static Block
- ④ Backtracking

[해설]

알고리즘 설계 기법

- 분할 정복/분할 통치(Divide and Conquer)
- 동적 계획법(Dynamic Programming)
- 탐욕 알고리즘(Greedy Algorithm)
- 백트래킹(Backtracking)

58. EAI(Enterprise Application Integration)의 구축 유형으로 옳지 않은 것은?

- ① Point-to-Point
- ② Hub & Spoke
- ③ Message Bus
- ④ Tree

[해설]

EAI 구축 유형

Point-to-Point, Hub & Spoke, Message Bus(ESB), Hybrid

59. 인터페이스 구현 검증 도구가 아닌 것은?

- ① ESB
- ② xUnit
- ③ STAF
- ④ NTAF

[해설]

인터페이스 구현 검증 도구

xUnit, STAF, FitNesse, NTAF, Selenium, watir

60. 데이터 속성 간의 종속성에 대한 엄밀한 고려 없이 잘못 설계된 데이터베이스에서는 데이터 처리 연산 수행 시 각종 이상 현상이 발생할 수 있는데, 이러한 이상 현상이 아닌 것은?

- ① 검색 이상
- ② 삽입 이상
- ③ 삭제 이상
- ④ 갱신 이상

[해설]

이상(Anomaly)의 종류

- 삽입 이상(Insertion Anomaly)
- 삭제 이상(Deletion Anomaly)
- 갱신 이상(Update Anomaly)

61. 다음 중 단위 테스트를 통해 발견할 수 있는 오류가 아닌 것은?

- ① 알고리즘 오류에 따른 원치 않는 결과
- ② 탈출구가 없는 반복문의 사용
- ③ 모듈 간의 비정상적 상호 작용으로 인한 원치 않는 결과
- ④ 틀린 계산 수식에 의한 잘못된 결과

[해설]

단위 테스트로 발견 가능한 오류

- 알고리즘 오류에 따른 원치 않는 결과
- 탈출구가 없는 반복문의 사용
- 틀린 계산 수식에 의한 잘못된 결과

62. 물리데이터 저장소의 파티션 설계에서 파티션 유형으로 옳지 않은 것은?

- ① 범위 분할(Range Partitioning)
- ② 해시 분할(Hash Partitioning)
- ③ 조합 분할(Composite Partitioning)
- ④ 유닛 분할(Unit Partitioning)

[해설]

파티션 종류

범위 분할, 해시 분할, 조합 분할, 목록 분할, 라운드 로빈 분할

63. 해싱 함수(Hashing Function)의 종류가 아닌 것은?

- ① 제곱법(Mid-Square)
- ② 숫자 분석법(Digit Analysis)
- ③ 개방 주소법(Open Addressing)
- ④ 제산법(Division)

[해설]

해싱 함수의 종류

제산법, 제곱법, 폴딩법, 기수 변환법, 대수적 코딩법, 계수 분석법(숫자 분석법), 무작위법

64. 다음 중 클린 코드 작성 원칙으로 거리가 먼 것은?

- ① 누구든지 쉽게 이해하는 코드 작성
- ② 중복이 최대화된 코드 작성
- ③ 다른 모듈에 미치는 영향 최소화
- ④ 단순, 명료한 코드 작성

[해설]

클린 코드 작성 원칙

가독성, 단순성, 의존성 배제, 중복성 최소화, 추상화

65. 관계대수의 순수 관계 연산자가 아닌 것은?

- ① Select
- ② Cartesian Product
- ③ Division
- ④ Project

[해설]

순수 관계 연산자의 종류

- Select( $\sigma$ )
- Project( $\pi$ )
- Join( $\bowtie$ )
- Division( $\div$ )

66. 디지털 저작권 관리(DRM)의 기술 요소가 아닌 것은?

- ① 크랙 방지 기술
- ② 정책 관리 기술
- ③ 암호화 기술
- ④ 방화벽 기술

[해설]

디지털 저작권 관리의 기술 요소

암호화, 키 관리, 암호화 파일 생성, 식별 기술, 저작권 표현, 정책 관리, 크랙 방지, 인증

67. 데이터 모델에 표시해야 할 요소로 거리가 먼 것은?

- ① 논리적 데이터 구조
- ② 출력 구조
- ③ 연산
- ④ 제약 조건

[해설]

데이터 모델에 표시할 요소

- 구조(Structure)
- 연산(Operation)
- 제약 조건(Constraint)

68. 소프트웨어 재공학이 소프트웨어의 재개발에 비해 갖는 장점으로 거리가 먼 것은?

- ① 위험 부담 감소
- ② 비용 절감
- ③ 시스템 명세의 오류 억제
- ④ 개발 시간의 증가

[해설]

소프트웨어 재공학의 장점

소프트웨어 수명 연장, 기술 향상, 개발 기간 단축, 오류 감소, 비용 절감, 위험 부담 감소

69. 테스트 케이스에 일반적으로 포함되는 항목이 아닌 것은?

- ① 테스트 조건
- ② 테스트 데이터
- ③ 테스트 비용
- ④ 예상 결과

[해설]

테스트 케이스의 기본 항목

입력 데이터, 테스트 조건, 예상 결과 등

70. 교착상태가 발생할 수 있는 조건이 아닌 것은?

- ① Mutual exclusion
- ② Hold and wait
- ③ Non-preemption
- ④ Linear wait

[해설]

교착상태 발생의 필요 충분 조건

- 상호 배제(Mutual Exclusion)
- 점유와 대기(Hold and Wait)
- 비선점(Non-preemption)
- 환형 대기(Circular Wait)

71. 공통 모듈의 재사용 범위에 따른 분류가 아닌 것은?

- ① 컴포넌트 재사용
- ② 더미코드 재사용
- ③ 함수와 객체 재사용
- ④ 애플리케이션 재사용

[해설]

재사용 규모에 따른 분류

- 함수와 객체
- 컴포넌트
- 애플리케이션

72. C언어에서 비트 논리 연산자에 해당하지 않는 것은?

- ① ^
- ② ?
- ③ &
- ④ ~

[해설]

C언어의 비트 연산자

&, ^, |, ~, <<, >>

73. C언어에서 산술 연산자가 아닌 것은?

- ① %
- ② \*
- ③ /
- ④ =

[해설]

C언어의 산술 연산자

+, -, \*, /, %

74. 다음 중 bash 셸 스크립트에서 사용할 수 있는 제어문이 아닌 것은?

- ① if
- ② for
- ③ repeat\_do
- ④ while

[해설]

셸 스크립트에서 사용되는 제어문

- 반복문 : for, while, until 등
- 조건문 : if, case 등

75. Java에서 사용되는 출력 함수가 아닌 것은?

- ① System.out.print( )
- ② System.out.println( )
- ③ System.out.printing( )
- ④ System.out.printf( )

[해설]

Java의 대표적인 출력 메소드

print( ), println( ), printf( )

76. 자바에서 사용하는 접근 제어자의 종류가 아닌 것은?

- ① internal
- ② private
- ③ default
- ④ public

[해설]

JAVA의 접근 제한자

Public, Default, Private, Protected

77. PHP에서 사용 가능한 연산자가 아닌 것은?

- ① @
- ② #
- ③ < >
- ④ ===

[해설]

PHP에서 사용 가능한 연산자

+, -, \*, /, %, @, < >, =, ==, ===

78. 운영체제에서 커널의 기능이 아닌 것은?

- ① 프로세스 생성, 종료
- ② 사용자 인터페이스
- ③ 기억장치 할당, 회수
- ④ 파일 시스템 관리

[해설]

커널(Kernel)의 기능

- 프로세스(CPU 스케줄링) 관리
- 기억장치 관리
- 파일 관리
- 입·출력 관리
- 프로세스 간 통신
- 데이터 전송 및 변환

79. 운영체제를 기능에 따라 분류할 경우 제어 프로그램이 아닌 것은?

- ① 데이터 관리 프로그램
- ② 서비스 프로그램
- ③ 작업 제어 프로그램
- ④ 감시 프로그램

[해설]

운영체제의 구성

- 제어 프로그램 : 감시 프로그램, 작업 관리 프로그램, 데이터 관리 프로그램
- 처리 프로그램 : 언어 번역 프로그램, 서비스 프로그램

80. UNIX SHELL 환경 변수를 출력하는 명령어가 아닌 것은?

- ① **configenv**
- ② printenv
- ③ env
- ④ setenv

[해설]

UNIX/LINUX에서 모든 환경 변수와 값을 표시하는 명령어  
set, env, printenv, setenv

81. 스크립트 언어가 아닌 것은?

- ① PHP
- ② **Cobol**
- ③ Basic
- ④ Python

[해설]

스크립트 언어

- 서버용 : ASP, JSP, PHP, 파이썬(Python), 펄(Perl), 루비(Ruby) 등
- 클라이언트용 : 자바 스크립트(Java Script), VB 스크립트(Visual Basic Script) 등

82. SQL의 논리 연산자가 아닌 것은?

- ① AND
- ② **OTHER**
- ③ OR
- ④ NOT

[해설]

SQL의 논리 연산자  
AND, OR, NOT

83. SQL의 명령어를 DCL, DML, DDL로 구분할 경우, 다음 중 성격이 다른 하나는?

- ① CREATE
- ② **SELECT**
- ③ ALTER
- ④ DROP

[해설]

데이터베이스 언어

- DDL(데이터 정의어) : CREATE, ALTER, DROP
- DML(데이터 조작어) : SELECT, UPDATE, INSERT, DELETE
- DCL(데이터 제어어) : COMMIT, ROLLBACK, GRANT, REVOKE, SAVEPOINT



84. 프로세스 상태의 종류가 아닌 것은?

- ① Ready
- ② Running
- ③ Request
- ④ Exit

[해설]

프로세스 상태

- 제출(Submit)
- 접수(Hold)
- 준비(Ready)
- 실행(Run)
- 대기(Wait), 보류, 블록(Block)
- 종료(Terminated, Exit)

85. 오류 제어에 사용되는 자동 반복 요청 방식(ARQ)이 아닌 것은?

- ① Stop-and-wait ARQ
- ② Go-back-N ARO
- ③ Selective-Repeat ARQ
- ④ Non-Acknowledge ARQ

[해설]

자동 반복 요청(ARQ)의 종류

- Stop-and-wait ARQ
- Go-back-N ARO
- Selective-Repeat ARQ
- Adaptive ARQ

86. WAS(Web Application Server)가 아닌 것은?

- ① JEUS
- ② JVM
- ③ Tomcat
- ④ WebSphere

[해설]

웹 애플리케이션 서버(WAS)의 종류

Tomcat, GlassFish, JBoss, Jetty, JEUS, Resin, WebLogic, WebSphere 등

87. 다음 중 페이지 교체(Page Replacement) 알고리즘이 아닌 것은?

- ① FIFO(First-In-First-Out)
- ② LUF(Least Used First)
- ③ Optimal
- ④ LRU(Least Recently Used)

[해설]

페이지 교체 알고리즘

OPT, FIFO, LRU, LFU, NUR, SCR 등

88. 데이터웨어하우스의 기본적인 OLAP(On-Line Analytical Processing) 연산이 아닌 것은?

- ① Translate
- ② Roll-Up
- ③ Dicing
- ④ Drill-Down

**[해설]**

OLAP 연산의 종류

- Roll-up
- Drill-down
- Drill-through
- Drill-across
- Pivoting
- Slicing
- Dicing

89. IPv6의 주소 체계로 거리가 먼 것은?

- ① Unicast
- ② Anycast
- ③ Broadcast
- ④ Multicast

**[해설]**

IPv6의 주소 체계

- 유니캐스트(Unicast)
- 멀티캐스트(Multicast)
- 애니캐스트(Anycast)

90. 병행제어 기법의 종류가 아닌 것은?

- ① 로킹 기법
- ② 시분할 기법
- ③ 타임 스탬프 기법
- ④ 다중 버전 기법

**[해설]**

병행제어 기법의 종류

- 로킹(Locking)
- 타임 스탬프 순서(Time Stamp Ordering)
- 최적 병행수행(검증 기법, 확인 기법, 낙관적 기법)
- 다중 버전 기법

91. 개발 환경 구성을 위한 빌드(Build) 도구에 해당하지 않는 것은?

- ① Ant
- ② Kerberos
- ③ Maven
- ④ Gradle

[해설]

빌드 자동화 도구

Ant, Maven, Gradle, Make, Jenkins 등

92. OSI 7계층 중 데이터 링크 계층에 해당되는 프로토콜이 아닌 것은?

- ① HTTP
- ② HDLC
- ③ PPP
- ④ LLC

[해설]

데이터 링크 계층의 프로토콜

HDLC, LAPB, LLC, MAC, LAPD, PPP

93. 다음 중 SQL의 집계 함수(Aggregation Function)가 아닌 것은?

- ① AVG
- ② COUNT
- ③ SUM
- ④ CREATE

[해설]

그룹 함수의 종류

COUNT, SUM, AVG, MAX, MIN, STDDEV, VARIANCE

94. TCP/IP 프로토콜 중 전송 계층 프로토콜은?

- ① HTTP
- ② SMTP
- ③ FTP
- ④ TCP

[해설]

TCP/IP의 전송 계층 프로토콜

TCP, UDP, RTCP

95. 소프트웨어 개발 프레임워크를 적용할 경우 기대 효과로 거리가 먼 것은?

- ① 품질 보증
- ② 시스템 복잡도 증가
- ③ 개발 용이성
- ④ 변경 용이성

[해설]

소프트웨어 개발 프레임워크의 기대 효과

- 품질 보증
- 개발 용이성
- 변경 용이성
- 시스템 복잡도 감소

96. 블록 암호화 방식이 아닌 것은?

- ① DES
- ② RC4
- ③ AES
- ④ SEED

[해설]

개인키 암호화 기법의 종류

- 블록 암호화 방식 : DES, SEED, AES, ARIA
- 스트림 암호화 방식 : LFSR, RC4

97. CMM(Capability Maturity Model) 모델의 레벨로 옳지 않은 것은?

- ① 최적 단계
- ② 관리 단계
- ③ 계획 단계
- ④ 정의 단계

[해설]

CMMI의 소프트웨어 프로세스 성숙도 5단계

- 초기(Initial)
- 관리(Managed)
- 정의(Defined)
- 정량적 관리(Quantitatively Managed)
- 최적화(Optimizing)

98. 나선형(Spiral) 모형의 주요 태스크에 해당되지 않는 것은?

- ① 버전 관리
- ② 위험 분석
- ③ 개발
- ④ 평가

[해설]

나선형 모형의 주요 태스크

계획 수립, 위험 분석, 개발 및 검증, 고객 평가

99. 취약점 관리를 위해 일반적으로 수행하는 작업이 아닌 것은?

- ① 무결성 검사
- ② 응용 프로그램의 보안 설정 및 패치(Patch) 적용
- ③ 중단 프로세스 및 닫힌 포트 위주로 확인
- ④ 불필요한 서비스 및 악성 프로그램의 확인과 제거

[해설]

취약점 관리 방법

- 무결성 검사
- 보안 설정 및 패치 적용
- 동작중인 프로세스 및 열린 포트 확인
- 로그 분석
- SetUID 파일 검사 등

100. 정보 보안의 3요소에 해당하지 않는 것은?

- ① 기밀성
- ② 무결성
- ③ 가용성
- ④ 휘발성

[해설]

정보 보안의 3대 요소

- 기밀성(Confidentiality)
- 무결성(Integrity)
- 가용성(Availability)

101. CBD(Component Based Development) SW 개발 표준 산출물 중 분석 단계에 해당하는 것은?

- ① 클래스 설계서
- ② 통합시험 결과서
- ③ 프로그램 코드
- ④ 사용자 요구사항 정의서

[해설]

CBD SW개발 표준의 단계별 산출물

- 분석 : 사용자 요구사항 정의서, 유스케이스 명세서, 요구사항 추적표
- 설계 : 클래스 설계서, 사용자 인터페이스 설계서, 컴포넌트 설계서, 인터페이스 설계서, 아키텍처 설계서, 총괄시험 계획서, 시스템시험 시나리오, 엔티티 관계 모형 기술서, 데이터베이스 설계서, 통합 시험 시나리오, 단위시험 케이스, 데이터 전환 및 초기 데이터 설계서
- 구현 : 프로그램 코드, 단위시험 결과서, 데이터베이스 생성 스크립트
- 시험 : 통합시험 결과서, 시스템시험 결과서, 사용자 지침서, 운영자 지침서, 시스템 설치 결과서, 인수시험 시나리오, 인수시험 결과서

102. 정보 보안을 위한 접근통제 정책 종류에 해당하지 않는 것은?

- ① 임의적 접근 통제
- ② 데이터 전환 접근 통제
- ③ 강제적 접근 통제
- ④ 역할 기반 접근 통제

**[해설]**

접근통제 기술의 종류

- 임의 접근통제(DAC; Discretionary Access Control)
- 강제 접근통제(MAC; Mandatory Access Control)
- 역할기반 접근통제(RBAC; Role Based Access Control)

103. Secure OS의 보안 기능으로 거리가 먼 것은?

- ① 식별 및 인증
- ② 임의적 접근 통제
- ③ 고가용성 지원
- ④ 강제적 접근 통제

**[해설]**

Secure OS의 보안 기능

- 식별 및 인증
- 임의적 접근 통제
- 강제적 접근 통제
- 객체 재사용 보호
- 완전한 조정
- 신뢰 경로
- 감사 및 감사기록 축소

104. 기능 점수(Functional Point) 모형에서 비용 산정에 이용되는 요소가 아닌 것은?

- ① 클래스 인터페이스
- ② 명령어(사용자 질의수)
- ③ 데이터 파일
- ④ 출력 보고서

**[해설]**

기능 점수(FP) 모형의 비용 산정 요인

- 자료 입력(입력 양식)
- 정보 출력(출력 보고서)
- 명령어(사용자 질의수)
- 데이터 파일
- 필요한 외부 루틴과의 인터페이스

105. 다음 암호 알고리즘 중 성격이 다른 하나는?

- ① MD4
- ② MD5
- ③ SHA-1
- ④ AES

[해설]

해시(Hash) 알고리즘의 종류

SHA 시리즈, MD4, MD5, N-NASH, SNEFRU 등

106. 상향식 비용 산정 기법 중 LOC(원시 코드 라인 수) 기법에서 예측치를 구하기 위해 사용하는 항목이 아닌 것은?

- ① 낙관치
- ② 기대치
- ③ 비관치
- ④ 모형치

[해설]

LOC 기법의 예측치를 구하는데 필요한 항목

비관치, 낙관치, 기대치

107. 세션 하이재킹을 탐지하는 방법으로 거리가 먼 것은?

- ① FTP SYN SEGMENT 탐지
- ② 비동기화 상태 탐지
- ③ ACK STORM 탐지
- ④ 패킷의 유실 및 재전송 증가 탐지

[해설]

세션 하이재킹 탐지 방법

- 비동기화 상태 탐지
- ACK Storm 탐지
- 패킷의 유실과 재전송 증가 탐지
- 예상치 못한 접속의 리셋 탐지

108. COCOMO 모델의 프로젝트 유형으로 거리가 먼 것은?

- ① Organic
- ② Semi-detached
- ③ Embedded
- ④ Sequential

[해설]

COCOMO의 소프트웨어 개발 유형

- 조직형(Organic Mode)
- 반분리형(Semi-Detached Mode)
- 내장형(Embedded Mode)

109. 익스트림 프로그래밍(eXtreme Programming)의 5가지 가치에 속하지 않는 것은?

- ① 의사소통
- ② 단순성
- ③ 피드백
- ④ 고객 배제

[해설]

XP의 5가지 핵심 가치

- 의사소통(Communication)
- 단순성(Simplicity)
- 용기(Courage)
- 존중(Respect)
- 피드백(Feedback)

110. 테일러링(Tailoring) 개발 방법론의 내부 기준에 해당하지 않는 것은?

- ① 납기/비용
- ② 기술 환경
- ③ 구성원 능력
- ④ 국제 표준 품질 기준

[해설]

테일러링 개발 방법론의 내부적 기준

- 목표 환경 : 시스템의 개발 환경과 유형
- 요구사항 : 개발, 운영, 유지보수 등
- 프로젝트 규모 : 비용, 인력, 기간 등
- 보유 기술 : 프로세스, 개발 방법론, 산출물, 구성원의 능력 등

111. 소프트웨어 비용 추정 모형(Estimation Models)이 아닌 것은?

- ① COCOMO
- ② Putnam
- ③ Function-Point
- ④ PERT

[해설]

소프트웨어 비용 산정 기법

- 하향식 산정 기법 : 전문가 감정 기법, 델파이 기법 등
- 상향식 산정 기법 : LOC(원시 코드 라인 수) 기법, 개발 단계별 인월수 기법, 수학적 산정 기법 등
- 수학적 산정 기법 : COCOMO 모형, Putnam 모형, 기능 점수(Function Point) 모형 등



112. 서비스 지향 아키텍처 기반 애플리케이션을 구성하는 층이 아닌 것은?

- ① 표현층
- ② 프로세스층
- ③ 제어 클래스층
- ④ 비즈니스층

[해설]

서비스 지향 아키텍처(SOA) 기반 애플리케이션 구성 계층

- 표현(Presentation) 계층
- 업무-프로세스(Biz-Process) 계층
- 서비스 중간(Service Intermediary) 계층
- 애플리케이션(Application) 계층
- 데이터 저장(Persistency) 계층

113. DDoS 공격과 연관이 있는 공격 방법은?

- ① Secure shell
- ② Tribe Flood Network
- ③ Nimda
- ④ Deadlock

[해설]

DDoS 공격의 종류

- Trin00
- TFN(Tribe Flooding Network)
- Stacheldraht

114. ISO 12207 표준의 기본 생명 주기의 주요 프로세스에 해당하지 않는 것은?

- ① 획득 프로세스
- ② 개발 프로세스
- ③ 성능평가 프로세스
- ④ 유지보수 프로세스

[해설]

ISO/IEC 12207

- 기본 생명 주기 프로세스 : 획득, 공급, 개발, 운영, 유지보수 프로세스
- 지원 생명 주기 프로세스 : 품질 보증, 검증, 확인, 활동 검토, 감사, 문서화, 형상 관리, 문제 해결 프로세스
- 조직 생명 주기 프로세스 : 관리, 기반 구조, 훈련, 개선 프로세스